An LDPCC decoding algorithm based on Bowman-Levin approximation —Comparison with BP and CCCP—

Masato Inoue, Miho Komiya and Yoshiyuki Kabashima
Department of Computational Intelligence and Systems Science,
Interdisciplinary Graduate School of Science and Engineering,
Tokyo Institute of Technology, Yokohama 226-8502, Japan
Email: inoue@sp.dis.titech.ac.jp, miho@sp.dis.titech.ac.jp, and kaba@dis.titech.ac.jp

***Abstract*— Belief propagation (BP) and the concave convex procedure (CCCP) are both methods that utilize the Bethe free energy as a cost function and solve information processing tasks. We have developed a new algorithm that also uses the Bethe free energy, but changes the roles of the master variables and the slave variables. This is called the Bowman-Levin (BL) approximation in the domain of statistical physics. When we applied the BL algorithm to decode the Gallager ensemble of short-length regular low-density parity check codes (LDPCC) over an additive white Gaussian noise (AWGN) channel, its average performance was somewhat better than that of either BP or CCCP. This implies that the BL algorithm can also be successfully applied to other problems to which BP or CCCP has already been applied.**

I. Introduction

Recently, various statistical inference algorithms have become of interest in the field of large-scale information processing. Belief propagation (BP) [1] and the concave convex procedure (CCCP) [2] are among the most effective of the methods which minimize the Bethe free energy [3], [4]. In the field of practical application (e.g., the problem of decoding low-density parity check code (LDPCC) [5], [6]), BP and CCCP have both been successfully applied [7].

However, they are not the only methods that minimize the Bethe free energy. In this paper, we focus on the method of Lagrange undetermined multipliers used by both BP and CCCP, and derive a new algorithm by exchanging the roles of master variables and slave variables. This approach, called Bowman-Levin (BL) approximation [8], is sometimes used in the field of statistical physics as a way to find an extremum (a saddle, local minimum, or local maximum) of the Bethe free energy.

II. Low density parity check code (LDPCC)

The LDPCC decoding problem can be handled within a Bayesian framework. The prior probability of the codes, consisting of N binary bits ($\boldsymbol{x} \in \{+1,-1\}^N$), is defined as

$$P(\boldsymbol{x}) \propto \prod_{\mu}^{M}\left(1+\prod_{l\in\boldsymbol{\mu}} x_l\right), \qquad (1)$$

where $\mu = 1, ..., M$ denotes the parity index and $\boldsymbol{\mu}$ denotes the set of node indices involved in the μ-th parity. Similarly, $l = 1, ..., N$ denotes the bit index and $\boldsymbol{l}$ denotes the set of parity indices linking to the l-th bit. $|\boldsymbol{\mu}|$ and $|\boldsymbol{l}|$ denote the degree of μ-th parity and the l-th bit, respectively. The proportion means the normalization of a probability function – i.e., the summation of the probability for all possible arguments $\boldsymbol{x}$ – should be 1.

We consider a noisy channel with additive white Gaussian noise (AWGN); i.e., the probability of the received codes $\boldsymbol{y}$ is defined as

$$P(\boldsymbol{y}|\boldsymbol{x}) \propto \prod_{l}^{N} \exp\left(-\frac{(y_l - x_l)^2}{2\sigma^2}\right), \qquad (2)$$

where σ^2 denotes the variance of the noise. The posterior probability of the sent code can then be expressed as

$$P(\boldsymbol{x}|\boldsymbol{y}) \propto \left[\prod_{\mu}^{M}\left(1+\prod_{l\in\boldsymbol{\mu}} x_l\right)\right]\left[\prod_{l}^{N} \exp\left(x_l \frac{y_l}{\sigma^2}\right)\right]. \qquad (3)$$

To infer the sent code $\boldsymbol{x}$ by $\boldsymbol{y}$, we employ the maximum posterior marginal (MPM) solution,

$$\hat{x}_l = \arg\max_{x_l} \sum_{\boldsymbol{x}_{\setminus l}} P(\boldsymbol{x}|\boldsymbol{y}), \qquad (4)$$

which minimizes the bit error rate. On the other hand, the maximum a posteriori (MAP) solution minimizes the block error rate,

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} P(\boldsymbol{x}|\boldsymbol{y}), \qquad (5)$$

but is generally difficult to determine because of the exponential calculation cost.

III. Bethe free energy

One purpose of the Bethe free energy approach is to determine a set of marginal probabilities of a given probability,

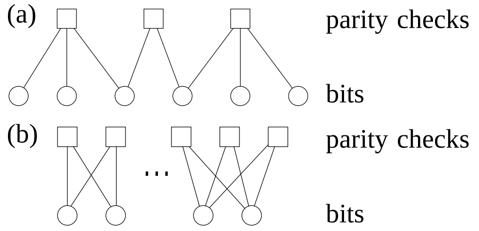

Fig. 1. Examples of the parity connection. (a) Tree structure and (b) loopy structure.

which provides the MPM solution here. The Bethe free energy, F, is defined using Kullback-Leibler (KL) divergence as

$$F \equiv \sum_{\mu}^{M} \mathrm{KL}(b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})||\phi_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})) \qquad (6)$$
$$+\sum_{l}^{N}(1-|\boldsymbol{l}|)\,\mathrm{KL}(q_l(x_l)||\psi_l(x_l)),$$

where $P(\boldsymbol{x}|\boldsymbol{y})$ can be represented as

$$P(\boldsymbol{x}|\boldsymbol{y}) \propto \left[\prod_{\mu}^{M}\phi_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})\right]\left[\prod_{l}^{N}\psi_l(x_l)^{1-|\boldsymbol{l}|}\right], \qquad (7)$$

$$\phi_\mu(\boldsymbol{x}_{\boldsymbol{\mu}}) \propto \left[1+\prod_{l\in\boldsymbol{\mu}} x_l\right]\left[\prod_{l}^{N}\psi_l(x_l)\right], \qquad (8)$$

$$\psi_l(x_l) \propto \exp\left(x_l\frac{y_l}{\sigma^2}\right), \qquad (9)$$

using (normalized) probability functions, $\phi_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})$ and $\psi_l(x_l)$ in the current case. The introduced test probability functions of creeks, $\{b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})\}$, and bits, $\{q_l(x_l)\}$, are required to satisfy the consistency of the marginal probabilities:

$$\sum_{\boldsymbol{x}_{\boldsymbol{\mu}\setminus l}} b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}}) = q_l(x_l) \qquad (l \in \boldsymbol{\mu}). \qquad (10)$$

$\{b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})\}$ and $\{q_l(x_l)\}$ that minimize F are expected to approximate the marginal probabilities of $P(\boldsymbol{x}|\boldsymbol{y})$.

The Bethe free energy approach gives the exact marginal probabilities when the parity connection has the tree structure (Fig. 1(a)). In such cases, any probability of $\boldsymbol{x}$ can be expressed as a product of its marginal probabilities, $\{b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})\}$ and $\{q_l(x_l)\}$, as

$$Q(\boldsymbol{x}) = \left[\prod_{\mu}^{M} b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})\right]\left[\prod_{l}^{N} q_l(x_l)^{1-|\boldsymbol{l}|}\right]. \qquad (11)$$

Then, the Bethe free energy coincides with the KL divergence between the test and the true probabilities,

$$F = \mathrm{KL}(Q(\boldsymbol{x})||P(\boldsymbol{x}|\boldsymbol{y})), \qquad (12)$$

which implies that minimizing the Bethe free energy leads to the correct probability $Q(\boldsymbol{x}) = P(\boldsymbol{x}|\boldsymbol{y})$, and, therefore, the exact MPM solution can be assessed from the obtained $\{q_l(x_l)\}$. Unfortunately, the Bethe free energy does not represent the KL divergence for loopy graphs (Fig. 1(b)). However, we here attempt to decode the LDPCC by minimizing F under the consistency condition (10) expecting that $\{q_l(x_l)\}$ well approximate the marginal probabilities even in the case that the parity connection does not have the tree structure.

IV. Lagrange multipliers

To minimize the Bethe free energy under the constraint (10), we introduce Lagrange undetermined multipliers, $\lambda_{\mu l}(x_l)$. The objective function to minimize is

$$G(\{b_\mu\},\{q_l\},\{\lambda_{\mu l}\}) \equiv F + L, \qquad (13)$$

where

$$L \equiv \sum_{\mu}^{M}\sum_{l\in\boldsymbol{\mu}}\sum_{x_l}\lambda_{\mu l}(x_l)\left(\sum_{\boldsymbol{x}_{\boldsymbol{\mu}\setminus l}} b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}}) - q_l(x_l)\right), \qquad (14)$$

and we solve the following three equations:

$$0 = \frac{\partial G}{\partial b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})} = \ln\frac{b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})}{\phi_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})} + 1 + \sum_{l\in\boldsymbol{\mu}}\lambda_{\mu l}(x_l), \qquad (15)$$

$$0 = \frac{\partial G}{\partial q_l(x_l)} = (1-|\boldsymbol{l}|)\left(\ln\frac{q_l(x_l)}{\psi_l(x_l)}+1\right) - \sum_{\mu\in\boldsymbol{l}}\lambda_{\mu l}(x_l), \qquad (16)$$

$$0 = \frac{\partial G}{\partial \lambda_{\mu l}(x_l)} = \sum_{\boldsymbol{x}_{\boldsymbol{\mu}\setminus l}} b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}}) - q_l(x_l). \qquad (17)$$

Using $x_l \in \{+1,-1\}$ and the normalization conditions of the probability functions, we can reduce q_l and $\lambda_{\mu l}$ to linear functions as

$$q_l(x_l) = \frac{1+x_l\tanh h_l}{2}, \qquad (18)$$

$$\lambda_{\mu l}(x_l) = -x_l\left(h_{\mu l}-\frac{y_l}{\sigma^2}\right) + \eta_{\mu l}. \qquad (19)$$

We also sometimes use $m_l \equiv \langle x_l\rangle_{q_l(x_l)} = \tanh h_l$. Using these expressions, we can reduce Eqs. (15) - (17) to

$$b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}}) \propto \left[1+\prod_{l\in\boldsymbol{\mu}} x_l\right]\left[\prod_{l}^{N}\exp(x_l h_{\mu l})\right], \qquad (20)$$

$$h_l = \frac{1}{|\boldsymbol{l}|-1}\left(\sum_{\mu'\in\boldsymbol{l}} h_{\mu' l} - \frac{y_l}{\sigma^2}\right), \qquad (21)$$

$$m_l = \frac{\sum_{x_l} x_l \sum_{\boldsymbol{x}_{\boldsymbol{\mu}\setminus l}} b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})}{\sum_{x_l}\sum_{\boldsymbol{x}_{\boldsymbol{\mu}\setminus l}} b_\mu(\boldsymbol{x}_{\boldsymbol{\mu}})}, \qquad (22)$$

respectively. From Eqs. (20) and (22), we obtain

$$h_l = h_{\mu l} + \mathrm{atanh}\prod_{l'\in\boldsymbol{\mu}\setminus l}\tanh h_{\mu l'} \qquad (23)$$

Now, we have two types of variable: $\{h_l\}$ and $\{h_{\mu l}\}$, and two types of simultaneous equation: (21) and (23).

V. Belief propagation (BP)

BP considers the double-indexed h, $\{h_{\mu l}\}$, to be the master variables. Specifically, from Eqs. (21) and (23), we obtain

$$h_{\mu' l} + \operatorname{atanh} \prod_{l' \in \boldsymbol{\mu}' \setminus l} \tanh h_{\mu' l'} = \frac{1}{|\boldsymbol{l}|-1}\left(\sum_{\mu'' \in \boldsymbol{l}} h_{\mu'' l} - \frac{y_l}{\sigma^2}\right) \tag{24}$$

for any $\{l, \mu' \in \boldsymbol{l}\}$. BP ingeniously rearranges the left side of this equation with the average without μ:

$$\frac{1}{|\boldsymbol{l}|-1} \sum_{\mu' \in \boldsymbol{l} \setminus \mu} \left(h_{\mu' l} + \operatorname{atanh} \prod_{l' \in \boldsymbol{\mu}' \setminus l} \tanh h_{\mu' l'} \right) = \frac{1}{|\boldsymbol{l}|-1}\left(\sum_{\mu'' \in \boldsymbol{l}} h_{\mu'' l} - \frac{y_l}{\sigma^2}\right) \tag{25}$$

We then obtain the iterative substitution to converge $\{h_{\mu l}\}$.

$$\text{loop: } h_{\mu l} \leftarrow \frac{y_l}{\sigma^2} + \sum_{\mu' \in \boldsymbol{l} \setminus \mu} \operatorname{atanh} \prod_{l' \in \boldsymbol{\mu}' \setminus l} \tanh h_{\mu' l'}. \tag{26}$$

Once the master variables are determined, we can easily obtain the slave variables, $\{h_l\}$, by

$$\text{result: } h_l \leftarrow \frac{y_l}{\sigma^2} + \sum_{\mu' \in \boldsymbol{l}} \operatorname{atanh} \prod_{l' \in \boldsymbol{\mu}' \setminus l} \tanh h_{\mu' l'}. \tag{27}$$

To lower the calculation cost, we check whether the estimated sent code, $\hat{x}_l \equiv \operatorname{sign} h_l$, satisfies all parities for every loop of Eq. (26). We stop the iteration loop if we reach any codeword, or the number of loops reaches an upper limit.

VI. Concave convex procedure (CCCP)

CCCP is a double loop algorithm utilizing convex optimization. The convexity of the Bethe free energy is generally not guaranteed because of the negative coefficient, $1-|\boldsymbol{l}|$, of the second term in Eq. (6). So, CCCP employs the following additional term at every outer loop step t.

$$\tilde{F}^t \equiv F + \sum_{l}^{N} |\boldsymbol{l}| \operatorname{KL}(q_l(x_l)||q_l^t(x_l)) \tag{28}$$

$$\begin{aligned} &= \sum_{\mu}^{M} \operatorname{KL}(b_\mu(\boldsymbol{x_\mu})||\phi_\mu(\boldsymbol{x_\mu})) + \sum_{l}^{N} \operatorname{KL}(q_l(x_l)||\psi_l(x_l)) \\ &\quad + \sum_{l}^{N} |\boldsymbol{l}| q_l(x_l) \ln \frac{\psi_l(x_l)}{q_l^t(x_l)}. \end{aligned} \tag{29}$$

Equation (29) guarantees the convexity of $\tilde{F}^t(\{b_\mu\},\{q_l\})$, because KL divergence function is convex, and the third term is a linear function. Besides, F necessarily decreases if $\tilde{F}^t$ decreases because the additional term is non-negative, and the additional term itself disappears if $\{q_l\}$ converges.

In the inner loop, similar to BP, CCCP considers the double-indexed h, $\{h_{\mu l}\}$, to be the master variables. On the other hand, in the outer loop, single-indexed h, $\{h_l\}$, are treated as the master variables. After the convergence of the inner loop, the outer loop is performed to determine h_l^{t+1}.

$$\begin{aligned} \text{inner loop: } h_{\mu l} \leftarrow \frac{1}{2} \Bigg(& \frac{y_l}{\sigma^2} + \sum_{\mu' \in \boldsymbol{l} \setminus \mu} (h_l^t - h_{\mu' l}) \\ & + h_l^t - \operatorname{atanh} \prod_{l' \in \boldsymbol{\mu} \setminus l} \tanh h_{\mu l'} \Bigg), \end{aligned} \tag{30}$$

$$\text{outer loop: } h_l^{t+1} \leftarrow \frac{y_l}{\sigma^2} + \sum_{\mu' \in \boldsymbol{l}} (h_l^t - h_{\mu' l}). \tag{31}$$

VII. Bowman-Levin (BL)

BL considers the single-indexed h, $\{h_l\}$, to be the master variables. Specifically, BL determines $\{h_{\mu l}\}$ by first using $\{h_l\}$ in Eq. (23). It requires some iteration to be solved, resulting in an inner loop:

$$\text{inner loop: } h_{\mu l} \leftarrow h_l^t - \operatorname{atanh} \prod_{l' \in \boldsymbol{\mu} \setminus l} \tanh h_{\mu l'}. \tag{32}$$

Because the determination of the slave variables, $\{h_{\mu l}\}$, depends on the provisional values of the master variables, $\{h_l\}$, BL also needs a double-loop algorithm. Eq. (21) implies that update

$$\text{outer loop: } h_l^{t+1} \leftarrow \frac{1}{|\boldsymbol{l}|-1}\left(\sum_{\mu' \in \boldsymbol{l}} h_{\mu' l} - \frac{y_l}{\sigma^2}\right), \tag{33}$$

may be employed for the outer-loop using the converged variables $\{h_{\mu l}\}$.

Eq. (33), however, does not provides satisfactory results as this empirically increases the Bethe free energy. This is because the outer loop (33) is interpreted as

$$h_l^{t+1} \leftarrow h_l^t + \kappa \frac{\partial G^t}{\partial h_l}. \tag{34}$$

where $\kappa \equiv \frac{\cosh^{-2} h_l^t}{|\boldsymbol{l}|-1}$ is positive. G^t denotes G, which is regarded as a function of only $\{h_l\}$ at outer-loop step t.

In order to resolve this difficulty, we use the natural gradient descent method [9] instead of Eq. (33) as

$$\text{outer loop: } \boldsymbol{h}^{t+1} \leftarrow \boldsymbol{h}_l^t - k\boldsymbol{H}^{-1} \frac{\partial G^t}{\partial \boldsymbol{h}}, \tag{35}$$

where k denotes a small positive step width, and $\boldsymbol{H}$ denotes the Fisher information matrix defined as

$$H_{i,j} \equiv \left\langle \frac{\partial \log Q(\boldsymbol{x})}{\partial h_i} \frac{\partial \log Q(\boldsymbol{x})}{\partial h_j} \right\rangle_{Q(\boldsymbol{x})} \tag{36}$$

$$= \begin{cases} \cosh^{-2} h_i & (i = j) \\ 0 & (i \neq j) \end{cases} \tag{37}$$

assuming the following approximation:

$$Q(\boldsymbol{x}) \simeq \prod_{l}^{N} q_l(x_l). \tag{38}$$

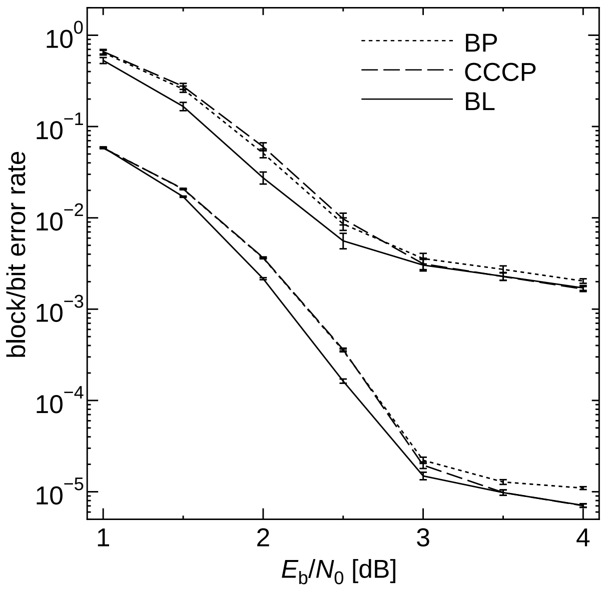

Fig. 2. Block (upper three lines) / bit (lower three lines) error rates of BP, CCCP, and BL algorithms. Configurations were as follows: code length $N = 486$, number of parities $M = 243$, degree of parity $|\boldsymbol{\mu}| = 6$, degree of bit $|\boldsymbol{l}| = 3$ ((3,6)-regular LDPCC), limit of outer loops: 10000 (the limit of loops in the case of BP), number of inner loops fixed as 6 for CCCP and BL, and step width $k = 0.3$ in BL. E_b/N_0[dB] is defined as $10\log_{10}(1/(2\sigma^2(N-M)/N))$. The number of communications were $10^3, 3\times 10^3, 10^4, 3\times 10^4, 10^5, 3\times 10^5$, and 10^6 for $E_\mathrm{b}/N_0 = 1.0, 1.5, \ldots, 4.0$, respectively. Each error bar denotes a 99% confidence interval based on a binomial distribution.

We then obtain

$$\text{outer loop: } h_l^{t+1} \leftarrow h_l^t - k\frac{\partial G^t}{\partial m_l}, \tag{39}$$

where

$$\frac{\partial G^t}{\partial m_l} = \sum_{\mu\in\boldsymbol{l}} h_{\mu l} - \frac{y_l}{\sigma^2} - (|\boldsymbol{l}|-1)h_l^t. \tag{40}$$

VIII. Validation

We validated the performance of the BL algorithm by comparing it with that of BP and CCCP through a simulation of the Gallager ensemble of the short-length regular LDPCC. As the decoding performance greatly depends on the parity check matrix, the simulation was done over an LDPCC ensemble; that is, we remade the matrix for every communication according to the Gallager's construction [5]. We assumed that the decoder knows the true noise variance of the AWGN channel, σ^2. In the simulation, BL performed somewhat better than both BP and CCCP.

Figure 2 shows the block and bit error rates of each algorithm over various signal-to-noise ratios (E_b/N_0). BL performed better than BP and CCCP, especially in the area where E_b/N_0 was around 2 dB. The error floor appeared in the area where E_b/N_0 was greater than about 2.5 dB. This error floor probably occurred due to the short loop of the parity check matrix.

Figure 3 shows the time (outer-loop step) evolution of the rate for both the not-correctly-decoded and wrongly-decoded cases. In the early outer-loop steps, BP tended to reach the correct codeword first, and then CCCP and BL followed. In the later steps, BL continuously improved the block error rate, while CCCP had little effect after about the 500-th step. The effect of BP was intermediate. The rates for the wrongly-decoded case were almost the same among the three algorithms. These results suggest that the BL algorithm will outperform BP and CCCP if we can afford a high calculation cost – for example, 1000 outer-loop steps.

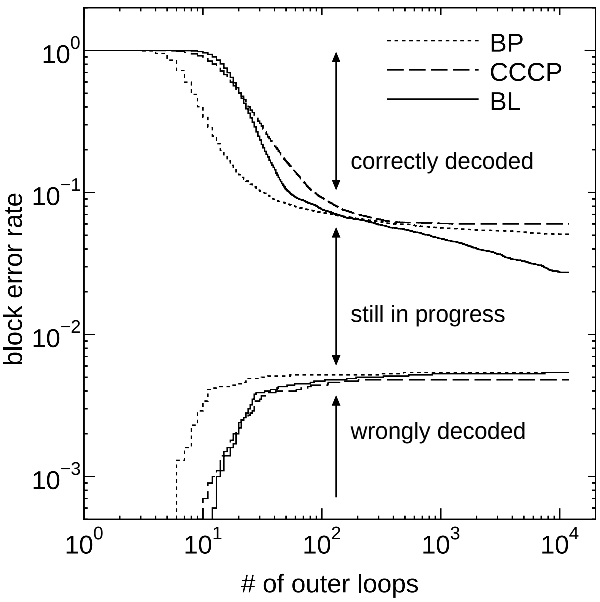

Fig. 3. Time evolution (outer loop steps) of the not-correctly-decoded rates (upper three lines) and wrongly-decoded rates (lower three lines) for BP, CCCP, and BL algorithms. The data corresponds to the case of $E_\mathrm{b}/N_0 = 2$ in Fig. 2.

The calculation cost is roughly proportional to the number of inner loops (we regard that of BP to be 1). So, if we set the number of inner loops as 6 for CCCP and BL, the cost ratio of BP, CCCP, and BL will be about $1 : 6 : 6$. If we consider the average number of outer loops, the difference could become larger (e.g., $1 : 10 : 12$), but this depends on the upper limit on the number of outer loops.

Parallelization is also an important factor regarding calculation cost. Briefly, parallelization of the BP loop is possible. It is also possible for the outer loops of CCCP and BL, but not for the inner loop of CCCP. On the other hand, it is indispensable for the inner loop of BL to achieve fast convergence.

Parameter optimization of the three algorithms is a real problem. In the case of BP, we have to determine only the upper limit of the outer loops. For CCCP, we also have to determine the number of inner loops. For BL, in addition to the CCCP parameters, we have to determine the step width of the natural gradient descent. Empirically, the configuration of the step width appears rather robust since the simulated BL performance generally exceeded that of the other algorithms (Fig. 2) even though they shared a common step width configuration (i.e., $k = 0.3$).

The optimization of the parity check matrix is also a problem, especially for short-length LDPCC. We will further investigate the dependence of these algorithms on the matrix in our future work.

IX. Conclusion

The method we have proposed minimizes the Bethe free energy based on the Bowman-Levin (BL) approximation. The BL algorithm combined with the natural gradient descent method successfully converges. We have compared our BL algorithm to the belief propagation (BP) and concave convex procedure (CCCP) algorithms with respect to the decoding problem of the Gallager ensemble of short-length regular low-density parity check codes (LDPCC) over an additive white Gaussian noise (AWGN) channel. Simulation showed that the BL algorithm outperformed the BP and CCCP algorithms, although the BL calculation cost was greater. This suggests that the BL algorithm can be successfully applied to other problems to which BP or CCCP have already been applied.

Acknowledgements

This work was partially supported by a Grant-in-Aid for Scientific Research on Priority Areas No. 14084206.

References

[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, 1988.
[2] A.L. Yuille, Neural Comput., **14**, 1691-1722, 2002.
[3] Y. Kabashima and D. Saad, Europhys. Lett., **44**, 668-674, 1998.
[4] J.S. Yedidia, W.T. Freeman and Y. Weiss, *Understanding Belief Propagation and its Generalization*, http://www.merl.com, MITSUBISHI RESEACH LABORATORIES.
[5] R. G. Gallager, *Low density parity check codes*, in Research Monograph series, Cambridge, MIT Press, 1963.
[6] D. J. C. MacKay, IEEE Trans. Inform. Theory, **IT-45**(2), 399-431, 1999.
[7] T. Shibuya and K. Sakaniwa, IEICE Trans. on Fundamentals, **E86-A**(10), 2601-2606, 2003.
[8] D. R. Bowman and K. Levin, Phys. Rev. B, **25**, 3438, 1982.
[9] S. Amari: Neural Comput. **10**, 251-276, 1998.